

OOP Introduction

January 7, 2020

1 Introduction to Object-Oriented Programming

1.1 Buzzword Compliance

- responsibility-driven design
- inheritance
- encapsulation
- overriding
- collection classes

1.2 Goals

- knowledge of programming principles
- sound knowledge of OOP
- critical assesement of small software systems
- design, build and test well-behaved objects

1.3 Classes

A category or type of thing (compare **struct** in C). A template or a blueprint. Unlike a **struct** in C, can contain both data and *instructions* that may be performed on the data stored in the class. The class should describe all objects of a particular kind in an *abstract* way.

1.4 Objects

An *object* belongs to a particular class and has individual characteristics. An instance of a class.

For example, for a class grading app, **Student** may be declared as a class. Objects **Alice**, **Bob**, **Charles** represent the individual students in the class. **Student.average()** could return the grade point average for a particular student.

1.5 Methods

We communicate with objects by invoking methods on them. An object is usually expected to do something when its methods are called.